Open access • Journal Article • DOI:10.1049/IP-G-1.1983.0036

Very fast discrete Fourier transform, using number theoretic transform

Wan-Chi Siu, Anthony G. Constantinides

Institutions: Imperial College London

Published on: 01 Oct 1983

Topics: Discrete Fourier transform (general), Discrete sine transform, Discrete Hartley transform, Split-radix FFT algorithm and Non-uniform discrete Fourier transform

Very fast discrete Fourier transform, using number theoretic transform

Wan-chi Siu, AP(HK), M.Phil., C.Eng., M.I.E.R.E., Mem.I.E.E.E., and
A.G. Constantinides, B.Sc(Eng.), Ph.D., C.Eng., M.I.E.E., Sen.Mem.I.E.E.E.

Indexing terms: *Mathematical techniques, Transforms*

Abstract: It is shown that number theoretic transforms (NTT) can be used to compute discrete Fourier transform (DFT) very efficiently. By noting some simple properties of number theory and the DFT, the total number of real multiplications for a length-P DFT is reduced to $(P-1)$. This requires less than one real multiplication per point. For a proper choice of transform length and NTT, the number of shift adds per point is approximately the same as the number of additions required for FFT algorithms.

1 Introduction

Direct computation of length-N discrete Fourier transform [1] (DFT) requires N^2 multiplications. The number of multiplications reduces to $\frac{1}{2}N \log_2 N$ if the fast Fourier transform algorithm [2] (FFT) is used. Winograd [3] showed that the minimum number of multiplications required to compute the circular convolution of two length-N sequences is $2N - K$, where K is the number of divisors of N including 1 and N. Agarwal and Cooley [4], Winograd [5] and Kolba and Parks [6] made use of Rader's theorem [7] on DFT with prime transform length to construct their algorithms for the computation of DFT. Compared to conventional FFT method, the Winograd Fourier transform algorithms reduce the number of multiplications by a factor of two to three, with a slightly large number of additions. Reed and Truong [8] proposed a technique for the computation of discrete Fourier transforms, based on Winograd's method in combination with Mersenne prime number-theoretic transforms. This hybrid algorithm requires fewer multiplications than either the standard FFT or Winograd's more conventional algorithm. However, a very large number of additions are required and the number of multiplications per point is still relatively large (about 1.33 to 3.49 multiplications per point). Recently, Nussbaumer [9] first defined a type of polynomial transform over the field of polynomials which could be used to compute 2-dimensional convolutions efficiently. This leads to the development of fast algorithms for the computation of multidimensional DFT [10, 11]. The number of multiplications in this case reduces to just two or three per point for sequences even longer than 1000 points.

In this paper, it is shown that the number of multiplications can be reduced further by using number theoretic transforms [12–15] to evaluate the DFT, and this forms a very efficient method of calculating the DFT.

2 Theory

Let the residue of the number g^n modulo P be written as $\langle g^n \rangle_P$, where g is a primitive root that generates all nonzero elements inside the field modulo P. Consider now an N-point discrete Fourier transform

$$Y(k) = \sum_{n=0}^{N-1} x(n) W_0^{nk} \tag{1}$$

where $k = 0, 1, \ldots, N-1$ and

$$W_0 = e^{-j(2\pi/N)}$$

If N is a prime number P, then eqn. 1 can be reordered [7] in the following form:

$$Y(0) = \sum_{n=0}^{P-1} x(n) \tag{2}$$

and

$$Y(\langle g^k \rangle_P) = x(0) + \sum_{n=1}^{P-1} x(\langle g^{-n} \rangle_P) W_0^{\langle g^{k-n} \rangle_P} \quad \text{for } k = 1, 2, \ldots, P-1 \tag{3}$$

We can write eqn. 3 as

$$Y(\langle g^k \rangle_P) = x(0) + X(\langle g^k \rangle_P) \tag{4}$$

where

$$X(\langle g^k \rangle_P) = \sum_{n=1}^{P-1} x(\langle g^{-n} \rangle_P) W_0^{\langle g^{k-n} \rangle_P} \quad \text{for } k = 1, 2, \ldots, P-1 \tag{5}$$

Eqn. 5 represents a backward circular convolution of length $(P-1)$. That is,

$$[x(g^{-1}), x(g^{-2}), \ldots, x(g^{-P+1})] \circledast [W_0^{g^0}, W_0^{g^1}, \ldots, W_0^{g^{P-2}}] \tag{6}$$

where $\circledast$ means circular convolution and the subscripts and indices are modulo P.

Let us now define

$$X_k = X(\langle g^{k+1} \rangle_P) \tag{7}$$

$$W_n = W_0^{\langle g^n \rangle_P} \tag{8}$$

$$x_n = x(\langle g^{-(n+1)} \rangle_P) \tag{9}$$

for $k = 0, 1, \ldots, P-2$; $n = 0, 1, \ldots, P-2$

Hence, eqns. 5 and 6 become

$$X_k = \sum_{n=0}^{P-2} x_n W_{k-n} \quad \text{for } k = 0, 1, \ldots, P-2 \tag{10}$$

and

$$(x_0, x_1, \ldots, x_{P-2}) \circledast (W_0, W_1, \ldots, W_{P-2}) \tag{11}$$

The number theoretic transform can now be applied to find the cyclic convolution sum of these two sequences. Thus we can write

$$X'_m = \left\langle \sum_{n=0}^{P-2} x_n \alpha^{mn} \right\rangle_M \tag{12}$$

Paper 2680G, first received 4th February and in revised form 6th May 1983

The authors are with the Department of Electrical Engineering, Imperial College of Science & Technology, London SW7 2BT, England. Mr. Siu is on leave from the Department of Electronic Engineering, Hong Kong Polytechnic, Hong Kong

and

$$W'_m = \left\langle \sum_{n=0}^{P-2} W_n \alpha^{mn} \right\rangle_M \quad \text{for } m = 0, 1, \ldots, P-2 \quad (13)$$

where α = a root of unity of order $(P-1)$ and M = base for modulo arithmetic.

The results can then be obtained by the inverse transform of the products, $X'_m W'_m$. That is,

$$X_k = \left\langle \frac{1}{P-1} \sum_{m=0}^{P-2} X'_m W'_m \alpha^{-mk} \right\rangle_M \quad \text{for } k = 0, 1, \ldots, P-2 \quad (14)$$

Recall that all W_ms are complex numbers; hence apparently the total number of multiplications for a real sequence of length P (to find all $X'_m W'_m$) for this method is $2(P-1)$. However, the sequence $(W_0^{g^0}, W_0^{g^1}, \ldots, W_0^{g^{P-2}})$ can actually be written as [5]

$$\{W_0^{g^0}, W_0^{g^1}, \ldots, W_0^{g^{[(P-1)/2]-1}}, W_0^{g^0}*, W_0^{g^1}*, \ldots W_0^{g^{[(P-1)/2]-1}}*\}$$

where * denotes complex conjugates.

Therefore, the sequence $(W_0, W_1, \ldots, W_{P-2})$ can be written as

$$\{W_0, W_1, \ldots, W_{[(P-1)/2]-1}, W_0^*, W_1^*, \ldots, W^*_{[(P-1)/2]-1}\}$$

Notice also that Real (W_n) = Real (W_n^*) and Imag (W_n) = $-$Imag (W_n^*). Hence,

$$\text{Real}\{W_{[n+(P-1)/2]}\} = \text{Real}(W_n) \quad (15)$$

$$\text{Imag}\{W_{[n+(P-1)/2]}\} = -\text{Imag}(W_n) \quad (16)$$

$$\text{for } n = 0, 1, \ldots, \left(\frac{P-1}{2} - 1\right)$$

In view of these relationships, eqn. 13 can be written as

$$W'_m = \left\langle \sum_{n=0}^{[(P-1)/2]-1} W_n \alpha^{mn} + \sum_{n=[(P-1)/2]}^{P-2} W_n \alpha^{mn} \right\rangle_M$$

$$= \left\langle \sum_{n=0}^{[(P-1)/2]-1} W_n \alpha^{mn} + \sum_{n=0}^{[(P-1)/2]-1} W_{[n+(P-1)/2]} \alpha^{[n+(P-1)/2]m} \right\rangle_M$$

and, since $\alpha^{[(P-1)/2]} = -1$, we can write

$$W'_m = \left\langle \sum_{n=0}^{[(P-1)/2]-1} W_n \alpha^{mn} + (-1)^m \sum_{n=0}^{[(P-1)/2]-1} W_{[n+(P-1)/2]} \alpha^{mn} \right\rangle_M \quad (17)$$

On combining eqns. 15–17, it is clear that

$$\text{Real}(W'_m) = \text{Real}\left(\left\langle 2 \sum_{n=0}^{[(P-1)/2]-1} W_n \alpha^{mn} \right\rangle_M\right) \quad \text{for } m = \text{even}$$

$$= 0 \quad \text{for } m = \text{odd} \quad (18)$$

$$\text{Imag}(W'_m) = 0 \quad \text{for } m = \text{even}$$

$$= \text{Imag}\left(\left\langle 2 \sum_{n=0}^{[(P-1)/2]-1} W_n \alpha^{mn} \right\rangle_M\right) \quad \text{for } m = \text{odd} \quad (19)$$

Eqns. 18 and 19 are very important in practical implementations, and this is not primarily because the number of shift (α multiplication if α is a simple combination of power of two) adds reduces by a factor of two for the calculation of W'_m, since all W'_m should be precalculated for hardware implementation; but because, however, the total number of real multiplications forming $X'_m W'_m$ reduces from $(2P-2)$ to $(P-1)$. This gives less than one [actually $1-(1/P)$] multiplication per point for the DFT of a real sequence of length P.

The number of additions required in this technique is evaluated below. The shift adds required by computing W'_m are not counted, since these quantities can be precalculated and stored in ROM for hardware implementation or stored in program for software implementation. The total number of shift adds required for transforming $(x_0, x_1, \ldots, x_{P-2})$ to $(X'_0, X'_1, \ldots, X'_{P-2})$ is $(P-1)(P-2)$. However, if we choose $(P-1)$ to be highly composite, an FFT-type algorithm can be applied to effect the transformation. In particular, if $(P-1)$ is a power of two, the number of shift adds is approximately equal to $(P-1)\log_2(P-1)$. Since the results are complex, two inverse transformations, one real and one imaginary, are required. Owing to the symmetry property of the DFT, only the first half of the length-$(P-1)$ inverse transform is necessary to compute. The other half of the inverse transform can be obtained by taking the conjugate of the first half of the inverse transform. Furthermore, both real and imaginary parts of the sequences $(X'_m W'_m, m = 0, 1, \ldots, P-2)$ are alternately zero, a length-$(P-1)$ inverse transformation can be formed by two length-$[(P-1)/2]$ inverse transforms. Hence, the number of shift-adds for the inverse transformation is

$$2\left(\frac{P-1}{2}\right)\left(\frac{P-1}{2} - 1\right) = (P-1)\left(\frac{P-1}{2} - 1\right)$$

in general, or is

$$2\left(\frac{P-1}{2}\right)\log_2\left(\frac{P-1}{2}\right) = (P-1)[\log_2(P-1) - 1]$$

if $(P-1)$ is a power of two. The total number of shift adds for $(P-1)$ being a power of two is $(P-1)[2\log_2(P-1) - 1]$. Therefore, the overall number of shift adds including the additions of $x(0)$ and the additions for $Y(0)$ becomes

$$(P-1)[2\log_2(P-1) - 1 + 2] = (P-1)[2\log_2(P-1) + 1]$$

This figure of shift adds is approximately equal to the number of real additions for FFT. Hence, the number of operations is significantly less than the number of operations reported in Reference 8.

3 Example

To illustrate the idea, let us consider the DFT of the sequence $[x(0), x(1), x(2), x(3), x(4)]$, i.e. $N = P = 5$. In this case, 2 is a primitive root which is used to generate elements inside the field modulo 5. Hence the mapping for $[x(n)]$ in eqns. 3 and 5 is given by

$$(x_n: n = 1, 2, 3, 4) = [x(\langle g^{-n}\rangle_M): n = 1, 2, 3, 4]$$

$$= [x(3), x(4), x(2), x(1)]$$

and

$$(W_0^{g^n}: n = 0, 1, 2, 3) = (W_0^1, W_0^2, W_0^4, W_0^3)$$

Hence, eqn. 6 becomes:

$$[x(3), x(4), x(2), x(1)] \circledast (W_0^1, W_0^2, W_0^4, W_0^3)$$

$$[x(3), x(4), x(2), x(1)] \circledast (W_0^1, W_0^2, W_0^{1*}, W_0^{2*}) \quad (20)$$

This convolution sum can be computed by NTT. Now let us use Fermat number transform (FNT) to make the calculation. Let $M = F_4 = 2^{16} + 1$, $\alpha = 2^8$ and, of course, $N = 4$. Hence,

$$\begin{bmatrix} X'_0 \\ X'_1 \\ X'_2 \\ X'_3 \end{bmatrix} = \left\langle \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 2^8 & -1 & -2^8 \\ 1 & -1 & 1 & -1 \\ 1 & -2^8 & -1 & 2^8 \end{bmatrix} \begin{bmatrix} x(3) \\ x(4) \\ x(2) \\ x(1) \end{bmatrix} \right\rangle_M \quad (21)$$

$$\begin{bmatrix} W'_0 \\ W'_1 \\ W'_2 \\ W'_3 \end{bmatrix} = \left\langle \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 2^8 & -1 & -2^8 \\ 1 & -1 & 1 & -1 \\ 1 & -2^8 & -1 & 2^8 \end{bmatrix} \begin{bmatrix} W_0^1 \\ W_0^2 \\ W_0^{1*} \\ W_0^{2*} \end{bmatrix} \right\rangle_M \quad (22)$$

where $W_0 = e^{-j(2\pi/5)}$.

In order to use modulo arithmetic, the W_0 terms have to be normalised to integer values. Multiplying these terms by 90 and rounding off the results to integers, we obtain:

$$\begin{bmatrix} W'_0 \\ W'_1 \\ W'_2 \\ W'_3 \end{bmatrix} = \begin{bmatrix} -90 + j0 \\ 0 + j38229 \\ 202 + j0 \\ 0 + j26964 \end{bmatrix}$$

This expression may be compared with eqns. 18 and 19 for agreement. Hence, for the computation of $X'_m W'_m$, $m = 0$, 1, 2, 3, a total number of four real multiplications is sufficient. This is also the total number of multiplications required for a 5-point DFT. The total number of real shift adds required is $4(2 \log_2 4 + 1) = 20$.

As we have seen, the length for the NTT is $(P - 1)$, where $(P - 1)$ is always an even number. Fermat number transforms [14], pseudo Mersenne transforms [16], pseudo Fermat transforms [17], or any efficient transform with even number of transform length, are suitable for the computation. However, for some very promising NTTs, the transform lengths may not be long enough or may not match this requirement. For example, an excellent choice of P is 257, which is prime, and $N(= P - 1)$ is highly composite, and it might be possible to use NTT to effect the convolution. The longest transform length (with $\sqrt{2}$ as the generator) for FNT with modulo base F_6 is 256. Hence F_6 is a possible choice for the implementation. If one wishes to use a shorter word-length, F_5 say, to make the implementation the major problem is that the maximum transform length for the FNT with $M = F_5 = 2^{32} - 1$ is 128 for $\alpha = \sqrt{2}$. However, this problem may be resolved by using multidimensional techniques for convolutions.* Since P is a prime number, it is also possible to combine Winograd's short DFTs to carry out the computation of long DFTs using multidimensional formulations [18]. The major disadvantages of the method using NTT to calculate DFT are that special arithmetic (modulo arithmetic) and normally relatively large word lengths may have to be used for the major part of the calculation—a fact common to all number theoretic transforms.

* The 1-dimensional 256-point cyclic convolution can be converted into a 2 × 128-point 2-dimensional convolution form. The 128-point cyclic convolution can be found by using FNT, whereas the other dimension is actually a length-2 linear convolution which can be computed by the Lagrange interpolation formula. If $(P - 1)$ is equal to the product of two mutually prime integers, the 1-dimensional cyclic convolution can be converted into a proper 2-dimensional cyclic convolution which can be computed by two number theoretic transforms or by a combination of the Winograd's convolution algorithm and the NTT.

4 Acknowledgment

The authors wish to express their gratitude to the referees for their helpful comments and recommendations.

5 References

1 CAPPELLINI, V., CONSTANTINIDES, A.G., and EMILIANI, P.: 'Digital filters and their applications' (Academic Press, 1978)

2 COOLEY, J.W., and TUKEY, J.W., 'An algorithm for the machine calculation of complex Fourier series', *Math. Comput.*, 1965, **19**, pp. 297–301

3 WINOGRAD, S.: 'Some bilinear forms whose multiplicative complexity depends on the field of constants', IBM res. rep. RC5669, IBM T.J. Watson Res. Ctr., 1975, NY, USA

4 AGARWAL, R.C., and COOLEY, J.W.: 'New algorithms for digital convolution', *IEEE Trans.*, 1977, **ASSP-25**, pp. 106–124

5 WINOGRAD, S.: 'On computing the discrete Fourier transform', *Math. Comput.*, 1978, **32**, pp. 175–199

6 KOLBA, D.P., and PARKS, T.W.: 'A prime factor FFT algorithm using high-speed convolution', *ibid.*, 1977, **ASSP-25**, pp. 91–103

7 RADER, C.M.: 'Discrete Fourier transforms when the number of data samples is prime', *IEEE Proc.*, 1968, **56**, pp. 1107–1108

8 REED, I.S., and TRUONG, T.K.: 'A new hybrid algorithm for computing a fast discrete Fourier transform', *IEEE Trans.*, 1979, **C-28**, pp. 487–492

9 NUSSBAUMER, H.J.: 'Digital filtering using polynomial transforms', *Electron. Lett.*, 1977, **13**, (13), pp. 386–387

10 NUSSBAUMER, H.J.: 'New polynomial transform algorithms for multidimensional DFTs and convolutions', *IEEE Trans.*, 1981, **ASSP-29**, pp. 74–83

11 NUSSBAUMER, H.J., and QUANDALLE, P.: 'Computation of convolution and discrete Fourier transforms by polynomial transforms', *IBM J. Res. & Dev.*, 1978, **22**, pp. 134–144

12 POLLARD, J.M.: 'The fast Fourier transform in a finite field', *Math. Comput.*, 1971, **25**, pp. 365–374

13 RADER, C.M.: 'Discrete convolution via Mersenne transform', *IEEE Trans.*, 1972, **C-21**, pp. 1269–1273

14 AGARWALL, R.C., and BURRUS, C.S.: 'Fast convolution using Fermat number transform with application to digital filtering', *ibid.*, 1974, **ASSP-22**, pp. 87–97

15 SIU, W.C.: 'Number theoretic transform and its applications to digital signal processing'. Proceedings of IERE, Hong Kong Section Workshop on Adv. Micro. and DSP, Hong Kong, Sept. 1982, pp. 76–101

16 NUSSBAUMER, H.J.: 'Digital filtering using complex Mersenne transforms', *IBM J. Res. & Develop.*, 1976, **20**, pp. 498–504

17 NUSSBAUMER, H.J.: 'Digital filtering using pseudo Fermat number transform', *IEEE Trans.*, 1977, **ASSP-26**, pp. 79–83

18 BURRUS, C.S.: 'Index mapping for multi-dimensional formulation of the DFT and convolution', *ibid.*, 1977, **ASSP-25**, pp. 239–242

Wan-chi Siu received the associateship of Hong Kong Polytechnic in electronic engineering in 1975 and the M.Phil. degree in electronics from the Chinese University of Hong Kong in 1977. From 1975 to 1980, he taught and subsequently became an electronic engineer in the Department of Electronics of the Chinese University of Hong Kong. Since 1980, he has been with the Department of Electronic Engineering of the Hong Kong Polytechnic as a lecturer. He is now on leave from Hong Kong Polytechnic and is with the Department of Electrical Engineering, Imperial College of Science and Technology, England. His research interests are in transform techniques, hardware and software implementations of digital signal processors, microprocessor architectures and fabrication technology. He is a chartered engineer and a member of the IERE and the IEEE.

A.G. Constantinides received the B.Sc.(Eng.) degree with first-class honours in 1965 and the Ph.D. degree from the University of London in 1968 for his research in digital filter design.

In 1969 he was an STL-sponsored research fellow at the City University, London, and later he became a Senior Research Fellow with the British Post Office Research Department. In 1971 he joined the Department of Electrical Engineering of the Imperial College of Science and Technology, London, where he is currently a Reader. He is a co-editor of the books 'Introduction of digital filtering', 'Digital signal processing' and co-author of the book 'Digital filters and their applications'.

Dr. Constantinides has been an active member of the IEEE, has served as vice-chairman of the Circuit Theory Chapter of the UK and Republic of Ireland, and on its Executive Committee. He has served as the first President of the European Association for Signal Processing (EURASIP) and is the co-chairman of the triennial International conference on digital signal processing (held in Florence, Italy). He is a chartered engineer and a member of the IEE.

Authors of Paper 2648G:

M.I. Sobhy received the B.Sc. degree in electrical engineering from the University of Cairo, Egypt, in 1956 and the Ph.D. degree from the University of Leeds, England in 1966.

He was a teaching assistant at the Department of Electrical Engineering, the University of Cairo until 1962, when he joined the University of Leeds, first as a research student and later as a lecturer working on microwave ferrite devices. In 1966 he joined Microwave Associates Ltd., Luton, England as a research engineer, where he worked on the development of microwave solid-state devices. He joined the University of Kent at Canterbury, England in 1967, where he is now leading a research group engaged on projects on computer-aided circuit deisgn, microwave circuits, digital filters and solid-state devices. He is also a consultant to a number of industrial establishments. Dr. Sobhy is a member of the IEE.

W. Surakampontorn was born in Bangkok, Thailand on 16th March 1952. He received his B.Eng. and M.Eng. degrees in electrical engineering from The King Mongkut's Institute of Technology, Ladkrabang Campus, Lad-krabang, Bangkok, Thailand, in 1976 and 1978, respectively.

He has been a member of the Faculty of Engineering at the King Mongkut's Institute of Technology, Ladkrabang, Bangkok, Thailand, since 1979, and is at present on study leave at the University of Kent, Canterbury, Kent, England. His research interests include digital signal processing and linear integrated circuit design.